```
#Calculate Wind Shear Data from Wind Profiler Data
#Code created by Ryan Ellis, WFO Raleigh, NC
#Date created 4/23/14


#Importing python packages matplotlib and numpy.

import matplotlib.pyplot as plt
import numpy as N
import os
from AllLLWS import profiledate

#deletes old profiledata file

os.remove('profiledata.txt')

print('i have started')
#open data files in current working directory and reads them in

# Enter the site id, pulse mode for the profiler and the date of the
data
profilerID = 'RST'
mode = '400'
filename= profilerID + '_' + mode + '.txt'
filename2='profiledata.txt'
#profiledate = '  ' + input('Enter the date in YY MM DD format:')
year=profiledate[2:4]
month=profiledate[5:7]
day=profiledate[8:10]

# Reads the file into a variable called f_lines
filename3='alltextrstresults/'+profilerID + '_' + year + month + day +
'shear.txt'
f = open(filename, 'r')
f_lines = f.readlines()
f.close()

f4 = open( filename3, 'w' )

# Set J and K indicies to zero and fill a string variable to look for
$ in data
#file

j=0;
k=0;

str1='$';

# looks for all hours on the date supplied and uses a 05 minutes to
get top of
```

```
# the hour data

for i in f_lines:
    if (i[0:10]== profiledate and i[14:16]=='29') or (i[0:10]==
profiledate and i[14:16]=='35') or (i[0:10]== profiledate and
i[14:16]=='36'):
        hr=i[11:13]

#this allows to skip the rest of the header
        k=j+7

#this looks for '$' to terminate reading data

        while f_lines[k].find(str1)==-1:
#creates a new file to write data to
            f2 =open(filename2, 'a+')
            a=f_lines[k]

#writes line to file and appends a new column with the hour in it
            f2.writelines(a.rstrip('\r\n') + '   ' + hr + '\r\n')

            k=k+1

# resets k and breaks if it finds $
            if f_lines[k].find(str1)!=-1:
                k=0
                break
    j=j+1

#closes the appendable data file

f2.close()

# re-opens the file as read only (only necessary if you want to look
at the data
# as the variable f3_lines

f3 = open('profiledata.txt', 'r')
f3_lines = f3.readlines()
f3.close()

#Using genfromtxt to load data from text file using a variable
whitespace
#delimiter and taking columns 0,1,2 and 12 for ht, wind speed,
direction, and
#hour. Then takes missing data that is listed as 999999 in the .txt
file
#and converts to NaN.

profiledata=N.genfromtxt('profiledata.txt',
```

```
                        delimiter=None,
                        usecols=(0,1,2,12),
                        missing_values='999999',
                        usemask=True)

#Creating arrays for variables the length of columns in the data file.
#Also defining variables

time=N.zeros(24)
ht0=N.zeros(24)
ws0=N.zeros(24)
wd0=N.zeros(24)



j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.124:
        ht0[j]=(profiledata[i,0])*3280 #converting km to ft
        time[j]=(profiledata[i,3])
        ws0[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws0[j]>50:
            ws0[j]=float('nan')
        wd0[j]=profiledata[i,2]
        j=j+1


ht1=N.zeros(24)
ws1=N.zeros(24)
wd1=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.17899999999999999:
        ht1[j]=(profiledata[i,0])*3280 #converting km to ft
        ws1[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws1[j]>50:
            ws1[j]=float('nan')
        wd1[j]=profiledata[i,2]
        j=j+1

ht2=N.zeros(24)
ws2=N.zeros(24)
wd2=N.zeros(24)

j=0

for i in range(len(profiledata)):
```

```
    if profiledata[i,0]==0.23300000000000001:
        ht2[j]=(profiledata[i,0])*3280 #converting km to ft
        ws2[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws2[j]>50:
            ws2[j]=float('nan')
        wd2[j]=profiledata[i,2]
        j=j+1

ht3=N.zeros(24)
ws3=N.zeros(24)
wd3=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.28799999999999998:
        ht3[j]=(profiledata[i,0])*3280 #converting km to ft
        ws3[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws3[j]>50:
            ws3[j]=float('nan')
        wd3[j]=profiledata[i,2]
        j=j+1

ht4=N.zeros(24)
ws4=N.zeros(24)
wd4=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.34300000000000003:
        ht4[j]=(profiledata[i,0])*3280 #converting km to ft
        ws4[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws4[j]>50:
            ws4[j]=float('nan')
        wd4[j]=profiledata[i,2]
        j=j+1

ht5=N.zeros(24)
ws5=N.zeros(24)
wd5=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.39800000000000002:
        ht5[j]=(profiledata[i,0])*3280 #converting km to ft
        ws5[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws5[j]>50:
            ws5[j]=float('nan')
```

```
        wd5[j]=profiledata[i,2]
        j=j+1

ht6=N.zeros(24)
ws6=N.zeros(24)
wd6=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.45300000000000001:
        ht6[j]=(profiledata[i,0])*3280 #converting km to ft
        ws6[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws6[j]>50:
            ws6[j]=float('nan')
        wd6[j]=profiledata[i,2]
        j=j+1

ht7=N.zeros(24)
ws7=N.zeros(24)
wd7=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.50800000000000001:
        ht7[j]=(profiledata[i,0])*3280 #converting km to ft
        ws7[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws7[j]>50:
            ws7[j]=float('nan')
        wd7[j]=profiledata[i,2]
        j=j+1

ht8=N.zeros(24)
ws8=N.zeros(24)
wd8=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.56299999999999994:
        ht8[j]=(profiledata[i,0])*3280 #converting km to ft
        ws8[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws8[j]>50:
            ws8[j]=float('nan')
        wd8[j]=profiledata[i,2]
        j=j+1

ht9=N.zeros(24)
ws9=N.zeros(24)
```

```
wd9=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.61799999999999999:
        ht9[j]=(profiledata[i,0])*3280 #converting km to ft
        ws9[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws9[j]>50:
            ws9[j]=float('nan')
        wd9[j]=profiledata[i,2]
        j=j+1

ht10=N.zeros(24)
ws10=N.zeros(24)
wd10=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.67300000000000004:
        ht10[j]=(profiledata[i,0])*3280 #converting km to ft
        ws10[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws10[j]>50:
            ws10[j]=float('nan')
        wd10[j]=profiledata[i,2]
        j=j+1

ht11=N.zeros(24)
ws11=N.zeros(24)
wd11=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.72799999999999998:
        ht11[j]=(profiledata[i,0])*3280 #converting km to ft
        ws11[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws11[j]>50:
            ws11[j]=float('nan')
        wd11[j]=profiledata[i,2]
        j=j+1

ht12=N.zeros(24)
ws12=N.zeros(24)
wd12=N.zeros(24)

j=0

for i in range(len(profiledata)):
```

```
    if profiledata[i,0]==0.78300000000000003:
        ht12[j]=(profiledata[i,0])*3280 #converting km to ft
        ws12[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws12[j]>50:
            ws12[j]=float('nan')
        wd12[j]=profiledata[i,2]
        j=j+1

ht13=N.zeros(24)
ws13=N.zeros(24)
wd13=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.83799999999999997:
        ht13[j]=(profiledata[i,0])*3280 #converting km to ft
        ws13[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws13[j]>50:
            ws13[j]=float('nan')
        wd13[j]=profiledata[i,2]
        j=j+1

ht14=N.zeros(24)
ws14=N.zeros(24)
wd14=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.89300000000000002:
        ht14[j]=(profiledata[i,0])*3280 #converting km to ft
        ws14[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws14[j]>50:
            ws14[j]=float('nan')
        wd14[j]=profiledata[i,2]
        j=j+1

ht15=N.zeros(24)
ws15=N.zeros(24)
wd15=N.zeros(24)

j=0

for i in range(len(profiledata)):
    if profiledata[i,0]==0.94799999999999995:
        ht15[j]=(profiledata[i,0])*3280 #converting km to ft
        ws15[j]=(profiledata[i,1])*1.94 #converting m/s to knots
        if ws15[j]>50:
            ws15[j]=float('nan')
```

```
        wd15[j]=profiledata[i,2]
        j=j+1

#Creating a figure with adjustable axes that can be manipulated later.
fig=plt.figure(figsize=(16.0, 5.0))

ax=fig.add_subplot(111)
ax.axis([-1,24,0,3500])
plt.xticks(N.arange(0, 24, 1.0))
plt.yticks(N.arange(0, 4000, 500))


k=0

for k in range(0,24):

ws=[ws0[k],ws1[k],ws2[k],ws3[k],ws4[k],ws5[k],ws6[k],ws7[k],ws8[k],ws9
[k],ws10[k],ws11[k],ws12[k],ws13[k],ws14[k],ws15[k]]

wd=[wd0[k],wd1[k],wd2[k],wd3[k],wd4[k],wd5[k],wd6[k],wd7[k],wd8[k],wd9
[k],wd10[k],wd11[k],wd12[k],wd13[k],wd14[k],wd15[k]]

ht=[ht0[k],ht1[k],ht2[k],ht3[k],ht4[k],ht5[k],ht6[k],ht7[k],ht8[k],ht9
[k],ht10[k],ht11[k],ht12[k],ht13[k],ht14[k],ht15[k]]

    llws=N.zeros(136);

    i=0


    for x in range(0,16):

        y=1
        while y < 16:

            if x is not y and x < y:

                llws[i]=N.sqrt(((ws[x])**2)+((ws[y])**2)-
(2*(ws[x])*(ws[y])*(N.cos(((N.deg2rad(wd[y])-(N.deg2rad(wd[x])))))))))
                llws[i]=N.around((llws[i]),decimals=1)
                y=y+1;
                i=i+1;

            else:

                y=y+1;


    a1=(llws[1], llws[16]);
    a2=(llws[2],llws[17],llws[31]);
```

```
    a3=(llws[3],llws[18],llws[32],llws[45]);
    a4=(llws[4],llws[19],llws[33],llws[46],llws[58]);
    a5=(llws[5],llws[20],llws[34],llws[47],llws[59],llws[70]);

a6=(llws[6],llws[21],llws[35],llws[48],llws[60],llws[71],llws[81]);

a7=(llws[7],llws[22],llws[36],llws[49],llws[61],llws[72],llws[82],llws
[91]);

a8=(llws[8],llws[23],llws[37],llws[50],llws[62],llws[73],llws[83],llws
[92],llws[100]);

a9=(llws[9],llws[24],llws[38],llws[51],llws[63],llws[74],llws[84],llws
[93],llws[101],llws[108]);

a10=(llws[10],llws[25],llws[39],llws[52],llws[64],llws[75],llws[85],ll
ws[94],llws[102],llws[109],llws[115]);

a11=(llws[11],llws[26],llws[40],llws[53],llws[65],llws[76],llws[86],ll
ws[95],llws[103],llws[110],llws[116],llws[121]);

a12=(llws[12],llws[27],llws[41],llws[54],llws[66],llws[77],llws[87],ll
ws[96],llws[104],llws[111],llws[117],llws[122],llws[126]);

a13=(llws[13],llws[28],llws[42],llws[55],llws[67],llws[78],llws[88],ll
ws[97],llws[105],llws[112],llws[118],llws[123],llws[127],llws[130]);

a14=(llws[14],llws[29],llws[43],llws[56],llws[68],llws[79],llws[89],ll
ws[98],llws[106],llws[113],llws[119],llws[124],llws[128],llws[131],llw
s[133]);

a15=(llws[15],llws[30],llws[44],llws[57],llws[69],llws[80],llws[90],ll
ws[99],llws[107],llws[114],llws[120],llws[125],llws[129],llws[132],llw
s[134],llws[135]);

    maxws=N.zeros(16);
    maxws[0]=llws[0];
    maxws[1]=N.nanmax(a1);
    maxws[2]=N.nanmax(a2);
    maxws[3]=N.nanmax(a3);
    maxws[4]=N.nanmax(a4);
    maxws[5]=N.nanmax(a5);
    maxws[6]=N.nanmax(a6);
    maxws[7]=N.nanmax(a7);
    maxws[8]=N.nanmax(a8);
    maxws[9]=N.nanmax(a9);
    maxws[10]=N.nanmax(a10);
    maxws[11]=N.nanmax(a11);
    maxws[12]=N.nanmax(a12);
    maxws[13]=N.nanmax(a13);
```

```
    maxws[14]=N.nanmax(a14);
    maxws[15]=N.nanmax(a15);


    vars()['maxws' + str(k)]=maxws


    for l in range(0,16):
        if  maxws[l] >= 30:
            ax.text(time[k], ht[l], maxws[l], color='red')
            f4.write("%2i, %7.2f, %6.1f\n" % (time[k], ht[l],
maxws[l]))
        elif maxws[l] >= 20 and maxws[l] <=30:
            ax.text(time[k], ht[l], maxws[l], color='orange')
            f4.write("%2i, %7.2f, %6.1f\n" % (time[k], ht[l],
maxws[l]))
        elif maxws[l] < 20:
            ax.text(time[k], ht[l], maxws[l], color='green')
            f4.write("%2i, %7.2f, %6.1f\n" % (time[k], ht[l],
maxws[l]))
        else:
            ax.text(time[k], ht[l], ' ') #this statement omits nans
from printing

f4.close()

ax.set_xlabel('Time (hr)')
ax.set_ylabel('Height (ft)')
ax.set_title(month+'/'+day+'/'+year+' ' + profilerID + ' Profiler Wind
Shear')

plt.grid()

plt.savefig('allrstresults/'+profilerID+'_'+year+month+day
+'_shear.png', dpi=None, facecolor='w', edgecolor='w',
        orientation='portrait')

#plt.show()
plt.close()
print(j)
```